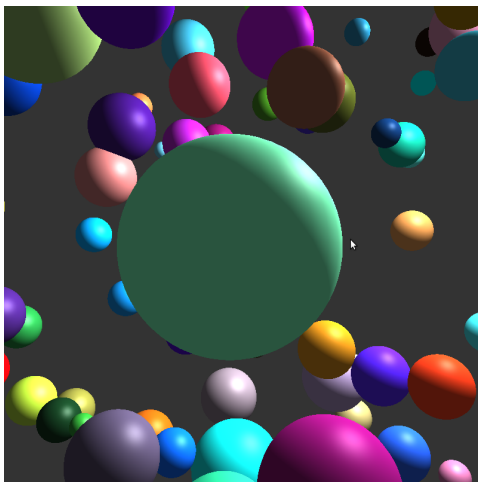


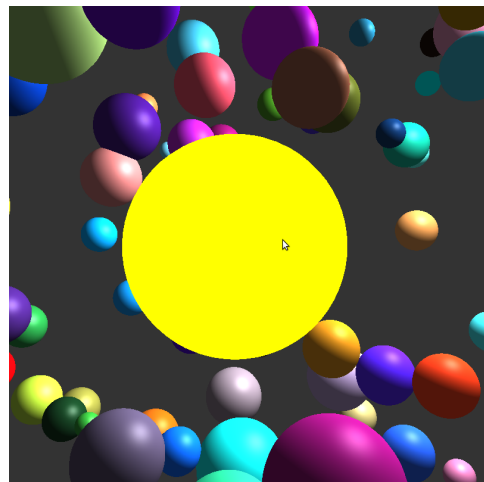


## Pointage d'objets

Le but de ce travail est de réaliser le pointage d'objets. Le pointage consiste à sélectionner un objet dans la scène 3D à l'aide de la souris. Cela revient à faire correspondre les coordonnées 2D de la souris à un objet 3D. Le pointage est notamment très utile pour interagir ou pour obtenir des informations sur un objet dans une scène complexe, constituée d'un grand nombre d'objets.



(a) Scène complexe avec un grand nombre d'objets



(b) Objet pointé avec la souris, mis en évidence en jaune

Ce travail sera effectué à l'aide de la bibliothèque Three.js. On rappelle que la bibliothèque peut être récupérée à l'adresse <https://raw.githubusercontent.com/mrdoob/three.js/master/build/three.min.js>

Une documentation très bien fournie est accessible à l'adresse : <https://threejs.org/docs/>.

Commencez par récupérer et lire le code contenu dans l'archive. Les parties à compléter se situent uniquement dans le fichier javascript et sont marquées d'un commentaire "TODO".

## 1 Création d'une scène complexe

Le but de cette partie est de créer une scène avec beaucoup d'objets. Vous devez créer une scène contenant 100 sphères, avec des couleurs variables et des rayons variables. Compléter la fonction `main()`. Voici les étapes :

1. Commencer par créer 100 sphères avec un rayon aléatoire compris entre 1 et 5 (utiliser la fonction `Math.random()`). La géométrie de la sphère est définie par `SphereGeometry`.
2. Changer la position des sphères de façon à ce qu'elles soient réparties aléatoirement dans un cube centré en  $[0, 0]$  et de côté égal à 100 (c'est-à-dire que les coordonnées  $x, y$  et  $z$  du centre des sphères sont comprises dans l'intervalle  $[-50, 50]$ ).
3. Donner une couleur aléatoire à chacune des sphères. La couleur est associée à un `MeshBasicMaterial`. Dans la suite de ce TP, ces couleurs sont appelées **couleurs réelles**.
4. Combiner matériau et géométrie dans un objet de type `Mesh`, et ajouter chaque instance de `Mesh` à la scène.
5. Ajouter une source de lumière dans la scène et faites en sorte que les sphères aient des ombres propres et des reflets (comme dans l'exemple montré dans la Figure (a)).

## 2 Implémentation du pointage d'objets

Le pointage d'objets se fait en attribuant une couleur **unique** à chacun des objets. La couleur pointée par la souris peut ensuite être retrouvée par la méthode `gl.readPixels()` du contexte WebGL. Cette méthode est déjà appelée dans la fonction `pointage()` du fichier javascript. La couleur aide ensuite à retrouver l'objet pointé.

L'objet sera retrouvé par son `id`, une variable qui associe la couleur récupérée par `gl.readPixels` à un indice dans le tableau où toutes les sphères sont stockées. Le tableau contenant les objets est défini en global et nommé `spheres`.

### 2.1 Attribution d'une couleur unique

**Préambule** Le pointage ne peut fonctionner que si on associe une couleur **unique** à une sphère. Or, les couleurs réelles ne sont pas nécessairement uniques (il est possible que deux sphères aient la même couleur). Vous allez attribuer une couleur différente à chaque sphère, de façon à créer un dégradé du noir (couleur de la première sphère créée) vers le bleu (couleur de la dernière sphère). Dans la boucle de création des sphères, voici le résultat de couleur attendu en fonction de l'itération :

Itération	0	1	...	255	256	257
Couleur (R,G,B)	0, 0, 0	0, 0, 1/255	...	0, 0, 255/255	0, 1/255, 0/255	0, 1/255, 1/255

**Question** Dans la boucle de création des sphères, créer un deuxième matériau avec `MeshBasicMaterial` contenant les couleurs uniques. **Indice** : utiliser l'opérateur modulo (%).

### 2.2 Stockage des objets dans un tableau

**Préambule** Il est nécessaire de stocker les sphères afin de pouvoir les retrouver lors du pointage par la souris. On souhaite aussi stocker les deux matériaux (couleurs réelles et uniques). Dans la boucle de création des sphères, créer des objets comportant ces 3 attributs :

- une instance de `Mesh`
- une instance du `MeshBasicMaterial` avec la couleur réelle
- une instance du `MeshBasicMaterial` avec la couleur unique.

L'objet peut être défini simplement par : `var obj = {nomAttrib1 : attrib1, nomAttrib2 : attrib2, nomAttrib3 : attrib3};`

**Question** Créer et stocker les objets dans le tableau `spheres`.

### 2.3 Rendu avec les deux matériaux

**Préambule** Pour effectuer le pointage, le rendu se fait en deux étapes :

1. On fait d'abord le rendu avec les couleurs uniques. A chaque couleur correspond un objet dans le tableau `spheres`. La couleur est récupérée avec la méthode `gl.readPixels()` et convertie en indice du tableau `spheres` (code fourni).
2. Puis on fait le rendu avec les couleurs réelles des sphères, c'est-à-dire celles définies dans la partie 1.

**Questions** Dans la fonction `pointage()` :

1. Redessiner les 100 sphères avec le matériau contenant la couleur unique (premier **TODO**)
2. Après récupération de l'indice, redessiner les 100 sphères avec les couleurs réelles des objets (deuxième **TODO**).
3. Si un objet est pointé par la souris, l'afficher en jaune (troisième **TODO**)

A l'issue de ces trois étapes, les couleurs uniques (dégradé de bleu) ne doivent pas apparaître à l'écran.